

Отчет

Домашнее задание №4

Выполнил: Иванов Д.Е. гр. М118

Задача: реализовать распределенный алгоритм умножения матрицы на вектор, рассмотреть зависимость времени, ускорения и эффективности от числа узлов, формы матрицы и мэппинга (для числа узлов 512).

Реализованы следующие утилиты (приведен формат вызова):

./main A.in B.in C.out – умножение матриц и запись в файл

./mul A.in B.in C.out– перемножение матриц (тупая последовательная реализация для сравнения)

./gen N M C.out – генерация матрицы в файл

./print C.in – вывод на печать в консоль матрицы из файла

./compare A.in B.in – сравнение матриц

Table 1. Время выполнения

rows	cols	1	32	64	128	256	512
512	512	0.005449	0.000176	0.00009	0.000047	0.000026	0.000017
1024	1024	0.021971	0.000712	0.000359	0.000183	0.000094	0.000047
1024	4096	0.09146	0.002992	0.001661	0.000999	0.000672	0.000493
2048	2048	0.089586	0.00278	0.001498	0.000763	0.000383	0.000196
4096	1024	0.087919	0.002749	0.001424	0.000714	0.000362	0.000173
4096	4096	0.365763	0.011439	0.006413	0.003209	0.001607	0.000806

Table 2. Ускорение

rows	cols	1	32	64	128	256	512
512	512	1	30.96023	60.54444	115.9362	209.5769	320.5294
1024	1024	1	30.85815	61.20056	120.0601	233.734	467.4681
1024	4096	1	30.56818	55.06321	91.55155	136.1012	185.5172
2048	2048	1	32.22518	59.80374	117.4128	233.906	457.0714
4096	1024	1	31.98218	61.74087	123.1359	242.8702	508.2023
4096	4096	1	31.97509	57.03462	113.9804	227.6061	453.8002

Table 3. Эффективность

rows	cols	1	32	64	128	256	512
512	512	1	0.967507	0.946007	0.905751	0.81866	0.626034
1024	1024	1	0.964317	0.956259	0.93797	0.913024	0.913024
1024	4096	1	0.955256	0.860363	0.715246	0.531645	0.362338
2048	2048	1	1.007037	0.934433	0.917288	0.913695	0.892718
4096	1024	1	0.999443	0.964701	0.961999	0.948712	0.992583
4096	4096	1	0.999221	0.891166	0.890472	0.889086	0.886329

Table 4. Время мэппинг

			Time		
Nodes	Rows	Cols	map1.txt	map2.txt	map3.txt
512	512	512	0.000017	0.000017	0.000017
512	1024	1024	0.000046	0.000047	0.000046
512	1024	4096	0.000484	0.000488	0.000487
512	2048	2048	0.000194	0.000193	0.000193
512	4096	1024	0.000174	0.000173	0.000174
512	4096	4096	0.000806	0.000806	0.000806

Table 5. Ускорение мэппинг

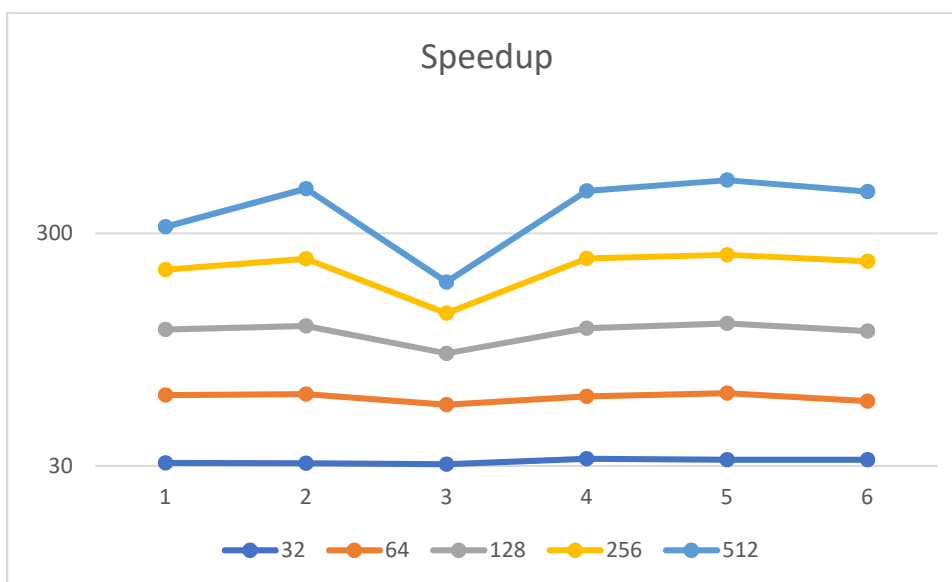
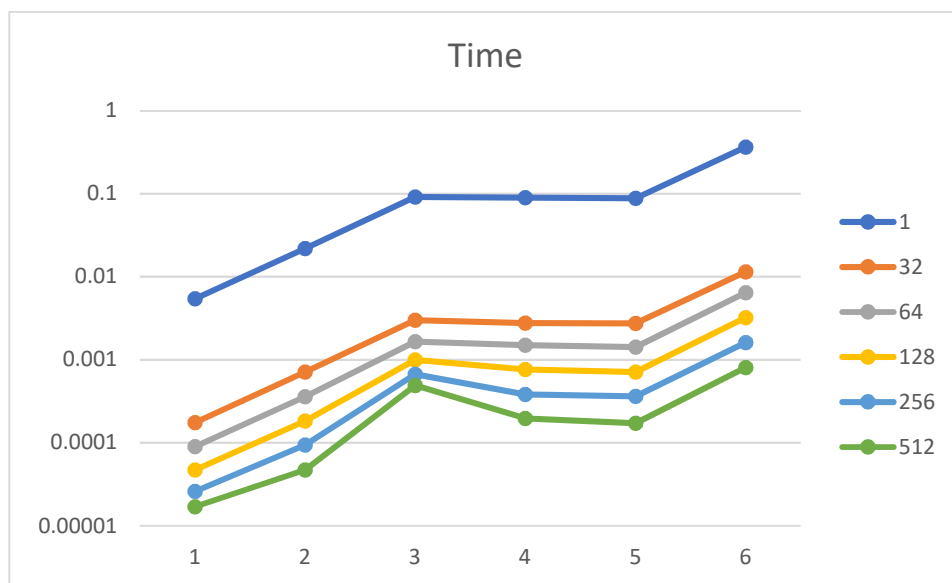
			Speedup		
Nodes	Rows	Cols	map1.txt	map2.txt	map3.txt
512	512	512	320.5294	320.5294	320.5294
512	1024	1024	477.6304	467.4681	477.6304
512	1024	4096	188.9669	187.418	187.8029
512	2048	2048	461.7835	464.1762	464.1762
512	4096	1024	505.2816	508.2023	505.2816
512	4096	4096	453.8002	453.8002	453.8002

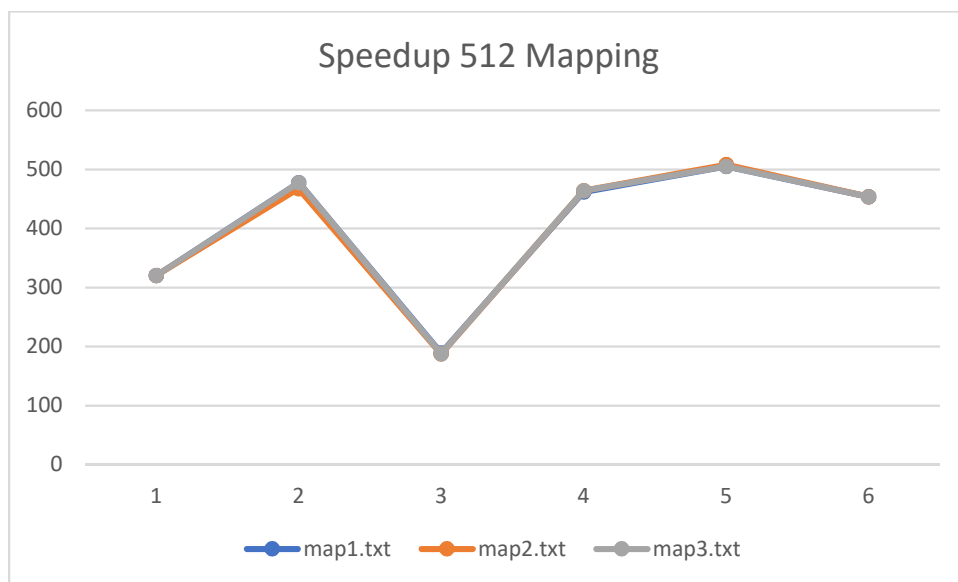
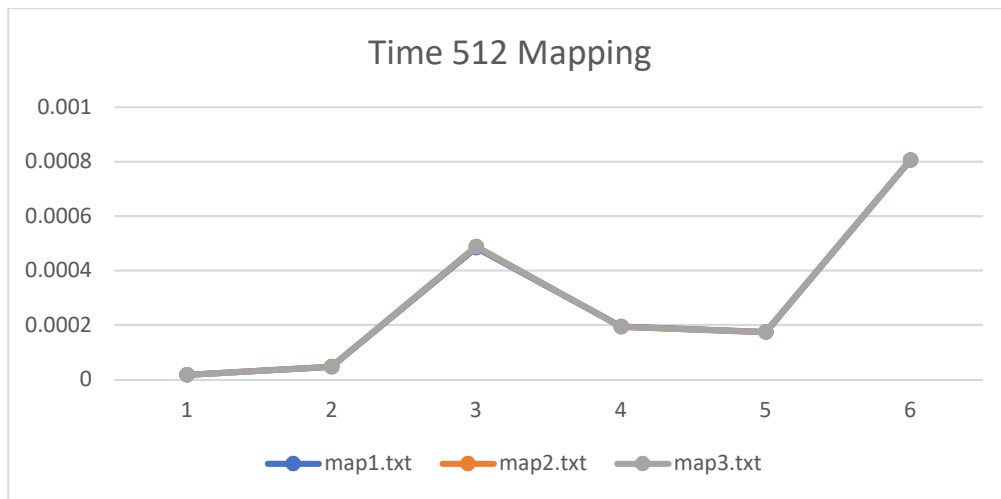
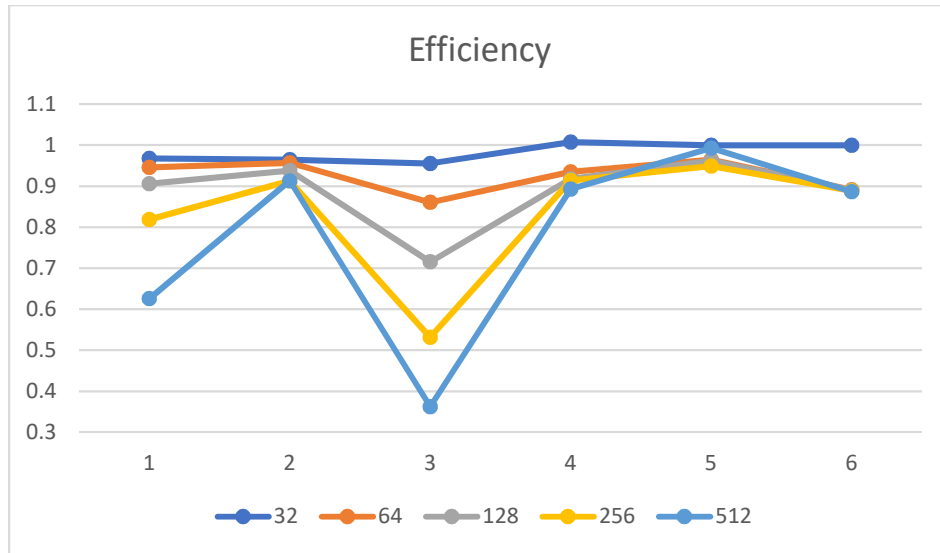
Table 6. Эффективность мэппинг

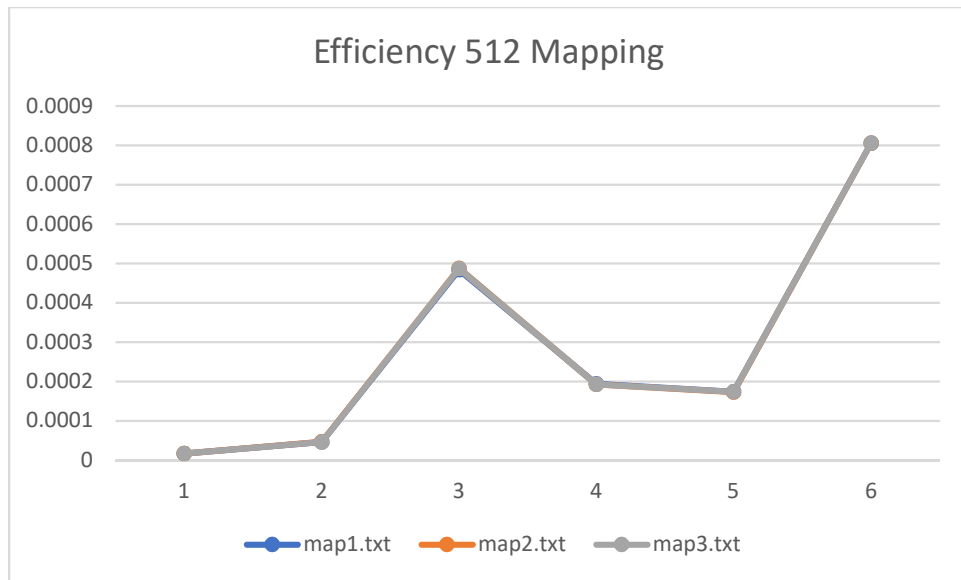
			Efficiency		
Nodes	Rows	Cols	map1.txt	map2.txt	map3.txt
512	512	512	0.000017	0.000017	0.000017
512	1024	1024	0.000046	0.000047	0.000046
512	1024	4096	0.000484	0.000488	0.000487
512	2048	2048	0.000194	0.000193	0.000193
512	4096	1024	0.000174	0.000173	0.000174
512	4096	4096	0.000806	0.000806	0.000806

На графиках ниже используются следующие обозначения

size_id	rows	cols
1	512	512
2	1024	1024
3	1024	4096
4	2048	2048
5	4096	1024
6	4096	4096







Вывод: оказывается, что эффективность применения алгоритма сильно зависит от соотношения сторон матрицы, а применение случайного мэппинга процессов почти не сказывается на канечном результате.